

# Funcionamiento de la Aplicación.

## Tecnologías:

- Angular 6
- Express (NodeJS)
- MongoDB
- HTML5
- Bootstrap 4

## Explicación Técnica:

Para generar el archivo JSON con el que se obtienen los datos para que angular pueda mostrarlos en pantalla o guardar en la Base de datos, se utilizó el Motor de Base de Datos no relacional MongoDB, se crearon dos API'S con el fin de poder manejar dos estructuras diferentes de archivos JSON, una de empleados y otra de pago.

A continuación, se da un ejemplo de la estructura de los archivos JSON:

JSON Empleados:

```
_id: "5b4545e1c0c52e3274b07d52"
cedula: 123
nombres: "pepito"
apellidos: "perez"
contrato: "Nº 76"
salario: 450
```

JSON Pago.

```
_id: "5b458d282b149152143c4db6"
cedula: "123"
fecha_inicial: "2018-07-01"
fecha_final: "2018-07-25"
bonificacion: 1750
```

Hay dos servicios que se comunican con el API que le corresponde a cada uno, dependiendo si el servicio es para tratar los datos de empleados o datos de pagos.

Los componentes de Empleados y Pagos se comunican con estos Servicios para hacer que la aplicación funcione como se espera.

Para correr el proyecto por consola se debe ir a la carpeta raíz del proyecto:

- [RUTA DEL SISTEMA]/nomina-app/.

Luego de que estemos allí, escribimos el siguiente comando:

- npm run dev.

Luego nos dirigimos a un explorador y escribimos la siguiente ruta:

- localhost:3000

## Explicación Funcional

En la primera pestaña que abre, esta el modulo de administración de empleados, en el se puede crear y eliminar un empleado, los campos que pide la aplicación para agregar un empleado son:

- Cédula
- Nombres
- Apellidos
- Número de Contrato
- Salario

En la parte de abajo hay una tabla en donde se puede ver los empleados que están registrados, al lado de cada empleado se muestran dos botones, uno es para eliminar el empleado y el otro es para ir a la parte de gestión del pago.

A continuación, se muestra una imagen de esta ventana.

## Listado de nomina

0



nombre

apellido

N°

0

Add employee

Contrato	Cédula	Nombres	Apellidos	Salario	Eliminar	Pagar
N° 5463	52646694	Pepito	Perez	3000000		

Cuando se dirige a la pestaña de pagos, esta cuenta con un formulario en el cual se pide:

- Cédula
- Fecha Inicial
- Fecha Fina
- Bonificación

En la parte de abajo hay una tabla en donde se puede ver los pagos asociados a una cédula que esten registrados, al lado de cada pago se muestran dos botones, uno es para eliminar el pago y el otro es para imprimir el comprobante de pago en formato PDF.

A continuación, se muestra una imagen de esta ventana.

# Listado de nomina



cedula

dd/mm/aaaa

dd/mm/aaaa

bonificacion

Add settlement

Cédula	Fecha Inicial	Fecha Final	Bonificacion	Borrar	Exportar PDF
25646897	2018-07-01	2018-07-15	1050000		

Empleados

## Funcionamiento de una parte del código

```
app-routing.module.ts
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { RouterModule, Routes } from '@angular/router';
4
5 //modules
6 import { PaymentComponent } from '../components/payment/payment.component';
7 import { SettlementComponent } from '../components/settlement/settlement.component';
8
9 const routes: Routes = [
10   {path: '', redirectTo: '/payment', pathMatch: 'full'},
11   {path: 'payment', component:PaymentComponent},
12   {path: 'settlement', component:SettlementComponent}
13 ];
14
15 @NgModule({
16   imports: [RouterModule.forRoot(routes)],
17   exports: [RouterModule]
18 })
19
20
21 export class AppRoutingModule { }
22
```

### App-routing-module.ts:

Esta clase nos ayuda a manejar el sistema de rutas entre componentes de nuestra aplicación.

Esto se utiliza cuando se desean que cada componente tenga una vista diferente dentro de nuestra aplicación.

¿Cómo Funciona?

Debemos importar las clases RouterModule y Routes de @angular/router y también cada componente que tengamos en la aplicación y que se desee tener en vistas diferentes.

Luego de ellos creamos un arreglo de Routes, con el fin de gestionar nuestro sistemas de rutas.

Posteriormente en el archivo (“../nombre\_proyecto\_angular/app/app.component.html/”) agregamos la siguiente línea

- <router-outlet></router-outlet>

## Retos Encontrados.

1. **Enrutamiento entre componentes en Angular:** Viendo a documentación de Angular logre solucionar esto.
2. **Manejar datos con archivo JSON:** Antes de realizar esta prueba siempre había usado Bases de datos relacionales, con lo que anteriormente no había necesitado generar una estructura JSON y manipular los datos que deseo almacenar o mostrar con esta estructura de archivos, buscar documentación de esto y ver que tecnología era la que más se adecuaba para esto fue lo que busque.
3. **Generar un PDF:** Lamentablemente esto no lo pude cumplir, puesto que busque como hacer esto, a la hora de implementarlo me generaba el PDF pero al momento de tratar de verlo no lo permitía porque decía que presentaba errores, es algo en lo que trabajare.

## Aportes a la Empresa si llego a ser colaborador

- Puedo aprender nuevas tecnologías con rapidez.
- Me gusta aprender nuevas cosas.
- Trabajo muy bien en equipo.
- Podría aportar a otras áreas diferentes al desarrollo, con algunas cosas que he aprendido de trabajos realizados.

Gracias,

Cristian Alejandro Wilches Ferreira.